

FIG. 1

200

280

Mapping Table for Unicode Conversion

Header: Fixed Identification Fields

282

**Unicode character corresponding to
source character with encoding 0**

284

**Unicode character corresponding to
source character with encoding 1**

286

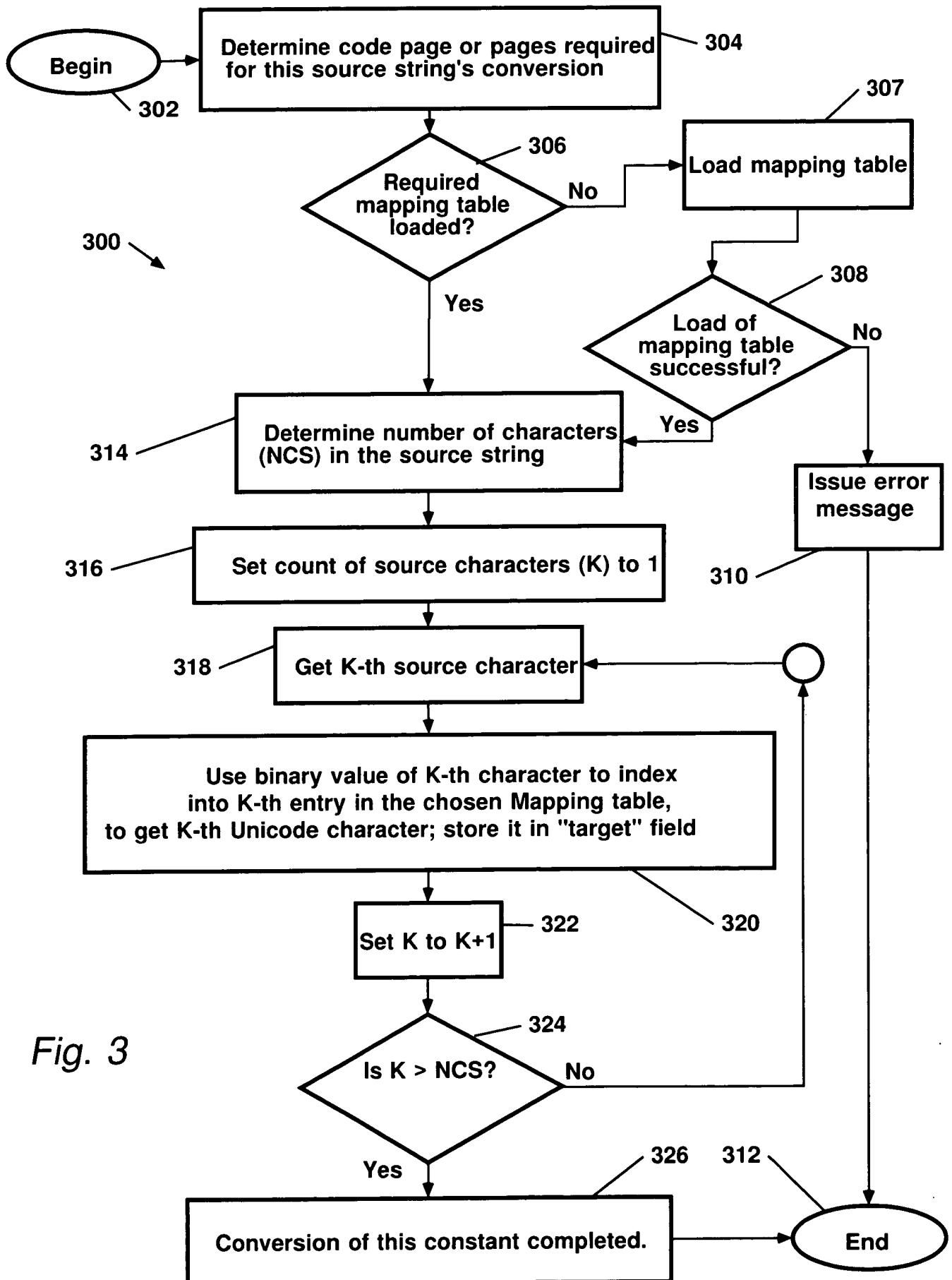
·
·
·

**Unicode character corresponding to
source character with encoding k**

288

·
·
·

Fig. 2



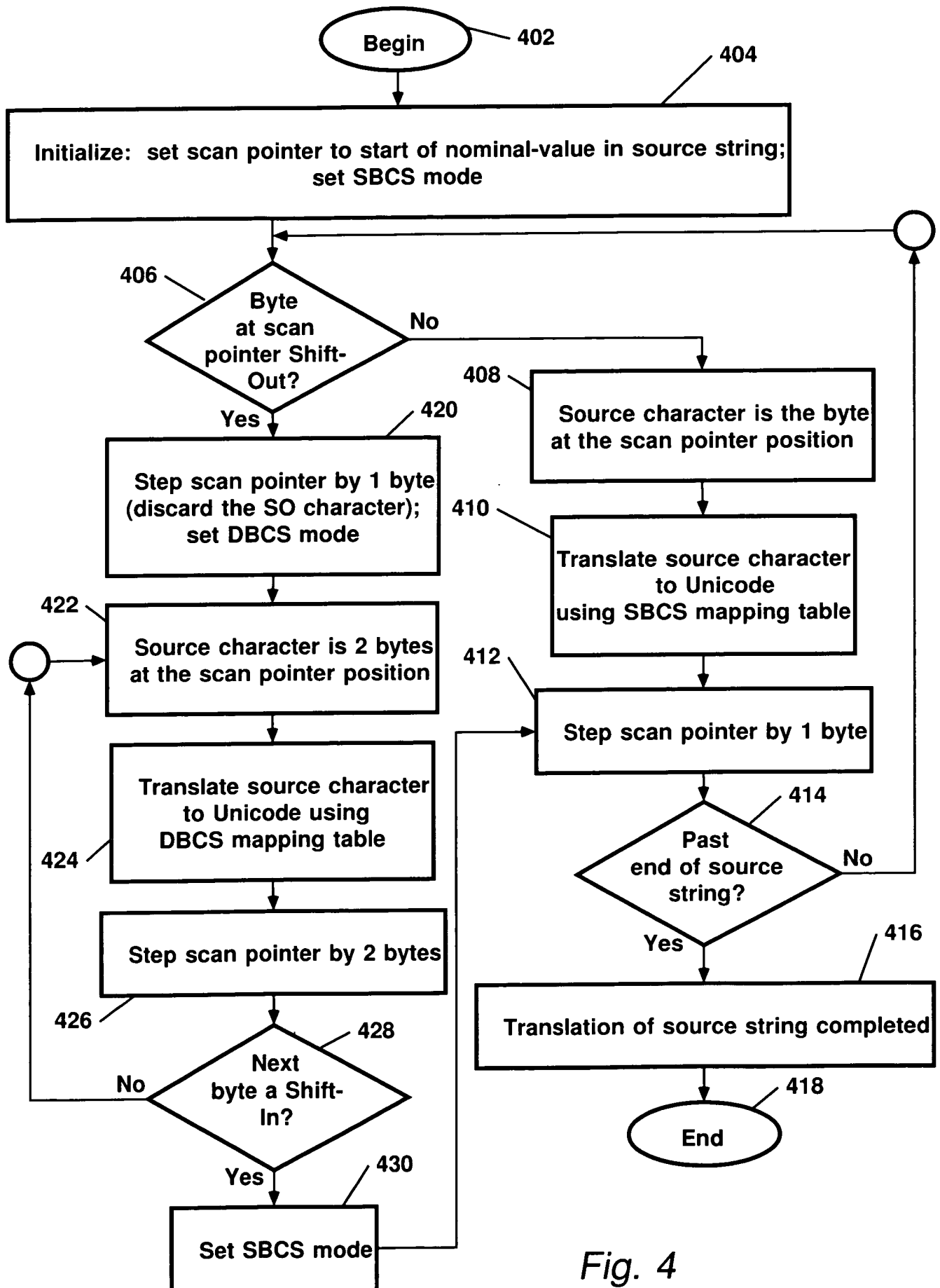


Fig. 4

Title 'DCU — a macro to generate Unicode constants'

Macro

```

&L   DCU   &A,&Pair=Yes,&CodePage=500
.*
.*
.*   Expected argument: an apostrophe-delimited string of one or
.*   more EBCDIC characters, with paired internal apostrophes and
.*   ampersands. The pairing is preserved in the output string if
.*   &Pair=Yes, and is not if &Pair=No.
.*   Initial limitation: max of 63 characters in quoted argument,
.*   except for paired characters if &Pair=No.
.*
.*-----*
.*   Declare variables used internally
.*-----*
.*
.*   LcIC &M(256)      Mapping and validation table
.*   LcIC &V           Valid EBCDIC characters
.*   LcIC &R           Result Unicode string
.*   LcIB &P           True if '&' pairs retained in output
.*   LcIA &J           Counter
.*   LcIA &N           Temp
.*-----*
.*   Validate macro arguments
.*-----*
.*
.*   Alf (N'&SysList gt 0).V1 Check for argument
.*   MNote 8,'DCU — No argument.'
.*   MExit
.*
.*.V1   Alf (N'&SysList lt 2).V2 Check single argument
.*   MNote 8,'DCU — More than one argument.'
.*   MExit
.*
.*.V2   Alf (K'&A ge 3).V3
.*   MNote 8,'DCU — argument too short, or badly formed.'
.*   MExit
.*
.*.V3   Alf ('&A'(1,1) eq "" and '&A'(K'&A,1) eq "").V4
.*   MNote 8,'DCU — argument not properly quoted.'
.*   MExit
.*
.*.V4   Alf ('YES' eq (Upper '&Pair')).V5 Check if pairing wanted
.*   Alf ('NO' eq (Upper '&Pair')).V6 Check if no pairing
.*   MNote 8,'DCU — invalid value of &&Pair.'
.*   MExit
.*
.*.V5   ANop ,
&P   SetB 1           Indicate no pairing of '&' in output
.*
.*

```

Fig. 5

```

.*
.V6  ANop ,
     Alf ('&CodePage' eq '500').V7  Check code page
     MNote 8,'DCU — Code Page &CodePage not supported yet.'
     MExit

.*
.V7  ANop ,
.*
.*  Arguments validated. Set SBCS and Unicode character sets  *
.*
.VX  ANop ,          Set up mapping table
&J  SetA &J+1
&M(&J) SetC "          Initialize to null
     Alf (&J lt 256).VX  Loop for all 256 code points

.*
&V  SetC '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz_@#%&*()-+=,./:;<>'?"' '

.*
.*  The following is the conversion table from CCSID 500 to Unicode
.*
&U  SetC '303132333435363738394142434445464748494A4B4C4D4E4F50515*
     2535455565758595A6162636465666768696A6B6C6D6E6F707172737*
     475767778797A5F4023242526262A28292D2B3D2C2E2F3A3B273C3E2*
     23F20'

.*
.*  Note: Conditional-assembly string constants require paired
.*  apostrophes and ampersands; ampersands are not reduced to a
.*  single character internally. Thus, the encoding for & appears
.*  twice in the &U encoding string above.
.*
&J  SetA 1
.*
.*  Build the EBCDIC-to-Unicode mapping table  *
.*
.VY  ANop
&C  SetC (Double '&V'(&J,1)) Pick character from valid string
&C  SetC 'C"&C"'          Character in self-defining term
&N  SetA &C+1              Convert to numeric
&M(&N) SetC '&U'(2*&J-1,2) Put Unicode digits in mapping table
&J  SetA &J+1              Increment &J
     Alf (&J le K'&V).VY  Set up all valid encodings

```

Fig. 6

```

.*
.*      Convert each SBCS argument character to Unicode equivalent      *
.*
&J      SetA 2                      Start after initial apostrophe
.*
.Z      ANop ,                      Head of translation loop
&C      SetC '&A'(&J,1)              &J-th character from argument
700 → Alf ('&C' ne "" and '&C' ne '&&(1,1)).Z1 Is it '& ?
Alf (&P).Z1                          Have '&, is pairing wanted?
&J      SetA &J+1                    No pairing, step input by one
.Z1     ANop ,
&C      SetC (Double '&C')          Pair '& for self-defining term
&C      SetC 'C"&C"'                Change to arithmetic value
&N      SetA &C+1                    Convert to numeric
&C      SetC '&M(&N)'               Get Unicode mapping
Alf ('&C' ne ").Z2 Validly mappable if not null
MNote 4,'DCU — Unknown character at position &J converted
          d to blank.'
&C      SetC '20'                    Unicode blank
.*
.Z2     ANop ,
&R      SetC '&R.00&C'              Add new character to end
&J      SetA &J+1
Alf (&J lt K'&A).Z Repeat for all internal characters
.*
.*      Generate the requested Unicode constant                        *
.*
&L      DC X'&R'
MEnd

```

Fig. 7